

Desenvolupament d'un joc tipus Pokémon-GO per a l'adquisició de mostres de text en escenes

Jordi Santamaría Portolés

Resum

Es vol fer servir la idea del pokemon Go de fer un joc on els usuaris tinguin que moures del seu lloc jugant, per a interactuar amb el seu entorn per a captar varietat de textos. D'aquesta forma aconseguim que el usuari s'exerciti una mica, al mateix temps que juga, además de que podem obtenir una base de dades de imatges on contingui text etiquetat amb la seva transcripció.

Aixó es útil per a la recerca, per a poder utilitzar aquestes imatges per a entrenar algorismes, sobretot de reconeixement de text. Actualment aquestes bases de dades de imatges es segueixen tenint diferents persones treballant en etiquetar-les manualment, el qual es una feina molt tediosa que es tarda molt de temps

Amb el nostre joc, aconseguiríem que el jugador, sense tenir coneixement previ sobre el tema, estigués ajudant a la recerca indirectament, al mateix temps que disfruta una bona estona jugant.

Per a desenvolupar aquest joc, s'ha proposat un joc basat en un laberint, en el qual per a poder-ne sortir, s'haurà de superar diferents proves reconeixent diferent textos amb la camera del mobil.

Paraules clau—Reconeixement de text en escenas, joc en android, computació humana fent ús de videojocs

Abstract

We had used the idea of Pokemon Go about to make a game where the users have to move from their location playing, for interactuate with the environment while capture a variety of texts. With this methode, we get the user to do a bit exercise, in same time he is playing, thus we can get a database with labeled images.

It's useful for a research, because we can use this images for train algorytms, especially about text recognition. Actually, this database of images are get, having some persons working in labeling manually, but it's very tedious and it takes a long time.

With our game, without the player be consciencious about it, he would be helping in research indirectly, while he get a funny time playing a while.

For desenvolupate this game, we propose a game based in a laberynt, where the player will have to overcome differents tests, recognizing a variety of words with the phone camera.

Index Terms— Reading in the wild, game in Android, Human Computation and purposeful gaming

1 INTRODUCCIÓ

Pokémon Go s'ha tornat tan popular que actualment és la paraula més buscada de Google¹. Una de les principals motivacions del seu èxit ha sigut gràcies al fet que

interactues amb el teu entorn mentre jugues, de forma que et permet aprofitar el fet de fer exercici, per poder actuar amb el joc, fent que sigui una activitat molt més amigable.

¹ <http://www.eltiempo.com/tecnosfera/novedades-tecnologia/pokemon-go-el-secreto-del-exito-del-videojuego-de-la-franquicia-de-nintendo/16646971>

Aquest fet d'aprofitar que l'usuari interactua amb la nostra aplicació, per aconseguir algun altre benefici, com en pokemon go seria el fet que la gent surti més al carrer i faci exercici, ja ha

sigut realitzat anteriorment en molts entorns, per exemple hi ha moltes pàgines web com facebook o amazon que es guarden dades dels usuaris per a futures recerques, però el èxit que ha tingut pokémon go, ha sigut la principal motivació que ens ha fet interessar-nos en aquest camp.

En el nostre cas, hem volgut aprofitar el fet de desenvolupar un joc, igual com ha fet pokémon Go, amb el qual els usuaris hagin de buscar paraules en el seu entorn, aconseguint que facin una mica d'exercici i no passant tant de temps assegut sense moure's, amb l'objectiu de poder reunir una base de dades amb una gran quantitat d'imatges que continguin texts de diferents orígens.

D'aquestes imatges, podem etiquetar quines paraules ha reconegut l'usuari i també podríem obtenir altra informació addicional d'utilitat com quina part de la imatge conté aquell text. Això és molt interessant d'aconseguir, donat que aquestes imatges, es podrien fer servir com a recerca per a poder millorar en gran mesura, com entrenar algorismes de reconeixement de text automàtics, donat que una de les majors limitacions per a que aquesta tecnologia de reconeixement de text avanci, és que el fet que la quantitat d'imatges etiquetades que hi han per recerca estan limitades.

Aquest joc que hem desenvolupat, l'hem fet per a la plataforma d'Android, s'ha decidit fer per a Android donat que així dona major facilitat a què els usuaris reuneixin imatges d'un conjunt d'entorns més variat que no pas una plataforma no portàtil.

Per a desenvolupar el joc, hem fet servir de la plataforma de Unity, juntament amb el plugin de Vuforia, el qual ens aporta la funcionalitat del reconeixement de text.

En aquest article, explicarem primer quins són els objectius que havíem decidit aconseguir, una mica el estat actual de les tecnologies que hem utilitzat, després explicarem la metodologia que s'ha seguit per aconseguir els nostres objectius, a continuació s'explicarà en què consisteix el nostre joc i com s'ha desenvolupat i per a concloure, mostrarem els nostres resultats, obtinguts al provar el joc amb diferents usuaris i acabarem amb una conclusió.

2 OBJECTIUS

En aquest projecte, ens basarem a crear un joc per a Android, fent ús de la realitat augmentada per a captar text en el món real.

Els nostres objectius esperats a aconseguir en finalitzar el projecte serien:

1. **Desenvolupar un joc en Android:**
multiplayer amb arquitectura client servidor.
2. **Integrar un mòdul de detecció de text:**
el suficient ràpid com perquè no afecti la experiència del usuari.

3. Evaluar el resultat final:

Un cop fet el joc, distribuir-lo a diferents usuaris, perquè evaluïn la seva experiència de joc, de forma que es puguin detectar i corregir possibles errors, com per exemple alguna falla en detectar el text, possibles millores en la interfaç del joc, la jugabilitat, etc..

3 ESTAT DE L'ART.

3.1 Reconeixent Text en escenas.

Introducció:

Per a saber l'estat actual del reconeixement de text, ens hem basat en las "ICDAR Robust Reading Competitions"[2], on la participació està oberta. La qual permet que els desenvolupadors puguin conèixer quins són els algorismes més eficients, tant com tenir una recollida d'imatges amb les quals basar-se per fer futures recerques.

Robust Reading es refereix al àrea d'investigació que s'ocupa de la interpretació de la comunicació escrita en entorns sense restriccions, típicament referenciat amb la detecció i reconeixement d'informació textual en imatges de escenes, però en un sentit més ampli, es refereix a tècniques i metodologies que s'han desenvolupat específicament per contenidors de text que no siguin paper escanejat o imatges d'origen digital o vídeos entre altres.

Tasks	Challenge 1: Born-Digital	Challenge 2: Focused Scene Text	Challenge 3: Text in Videos	Challenge 4: Incidental Scene Text
1. Localization / Tracking	2011 / 2013	2011 / 2013	2013 / 2015	2015
2. Segmentation	2011 / 2013	2013		
3. Recognition	2011 / 2013	2011 / 2013		2015
4. End-to-end	2015	2015	2015	2015

Figura 1: Tasques proposades en ICDAR

Aquesta competició ha sigut celebrada 5 vegades, en 2003, 2005, 2011, 2013 i 2015, al voltant de diferents dominis representatius de la lectura robusta, la qual es tornarà a celebrar de nou aquest any.

Tasques de Recerca: [3]

La competició està basada en 4 tasques:

- Tasca 1:** Localització de text: obtenir una estimació de àrees de text en la imatge.
- Tasca 2:** Segmentació del text: Separar el text del fons amb una tonalitat de pixel diferent.
- Tasca 3:** Reconeixement de les paraules: Obtenir la

transcripció correcta del text

-Tasca 4: D'extrem a extrem: Localitzar i reconèixer totes les paraules de la imatge.

Perquè els participants puguin competir amb els seus algorismes, es preparen unes bases de dades amb un conjunt d'imatges pel text i un altre per a l'entrenament, en el ICDAR 2015 es van adquirir 1600 imatges utilitzant google glass durant uns mesos.

En aquests conjunts d'imatges, s'ha etiquetat la seva informació important com les paraules que apareixen i en quines zones, per a poder avaluar el funcionament dels algorismes.

Resultats:

State of the art - 2015

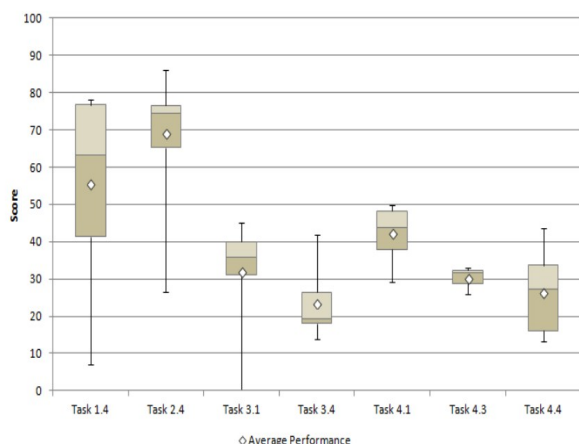


Figura 2: Resultats del ICDAR 2015

En la figura 2 podem veure els resultats que es van arribar en l'última competició, en aquests resultats ens podem donar compte que la tasca 1 i 2 provinents de localitzar i segmentar el text està bastant avançat, de forma que els algorismes s'equivocan molt poc, per altra banda la tasca 3 i 4 que provendria més en poder transcriure el text de la imatge, encara falta millorar bastant.

La tecnologia de reconeixement de caràcters OCR, està bastant avançada i funciona amb bastant bona precisió, per el fet de pasar-ho en escenes reals, és bastant més complex, algunes de les principals raons serien les següents: [4]

-Complexitat en escenes: Hi han molts objectes fets per l'home, tal com edificis, símbols, pintures, on la seva estructura i apariencia és similar al text, això fa que sigui complex discriminar el text del no text.

-Il·luminació desigual: La il·luminació fa que habituin distorsions en el color, cosa que genera falses deteccions.

-Distorsió i degradació: El qual es genera al captar fotos amb una càmera, depèn el seu enfocament el text pot aparèixer difuminat. També hi ha el factor extra de la compressió i

descompressió d'imatge/video, el qual degrada també la qualitat del text. Això genera que tasques bàsiques com la segmentació es dificulti.

-Relacions d'aspecte: El text pot ser més llarg o més gran depenen el context, això fa que es hagi de tenir en compte la localització, l'escala i la longitud, de forma que introdueix un augment de complexitat computacional.

-Distorsió: La distorsió de perspectiva succeeix quan l'eix de la càmera no està perpendicular al pla del text, tal com es pot veure a la figura 3.



Figura 3: Distorsió de la imatge

-Fonts: Caràcters en cursiva o en algunes fonts, poden solapar-se entre si, dificultant la segmentació.

-Entorns multilingües: Aunque la major part dels llenguatges provinents del llatí tenen desenes de caràcters, hi han llenguatges tals com el xines, el japonès i el corea, que tenen milers de caràcters.

3.2 Computació humana fent ús de videojocs

Hi ha molts casos de la utilització de videojocs per aconseguir objectius alternatius, però aprofitaré el cas den Luis von Ahn i la Laura Dabbish de la Universitat Carnegie Mellon, on van desenvolupar un joc per a etiquetar imatges [5], el qual es molt semblant al que volem aconseguir en el nostre joc, per explicar que s'ha arribat a aconseguir anteriorment amb aquesta tecnologia.



Figura 4: El juego ESP, Els jugadors tenen que intentar estar d'acord amb el màxim de imatges possibles en 2.5minuts. El termometre abaix mesura quantes imatges s'han completat

El joc que ells van desenvolupar, es basava de un joc multijugador entre 2 usuaris que no es podien comunicar entre ells, on tenien una imatge i havien de dir alguna cosa que els hi suggerís aquella imatge, havia un temps límit i tenien que aconseguir coincidir els 2 jugadors amb alguna de les paraules que suggerís per a poder passar a la següent imatge, l'objectiu era aconseguir completar el major número d'imatges en un temps concret. El joc contenia una primera versió de 350.000 imatges que van ser etiquetades amb les paraules que els usuaris havien anat coincidint, es van aconseguir tenir 5000 usuaris jugant el joc de forma constant durant 4 mesos, el qual es va aconseguir 1.271.451 etiquetes per a 293.760 imatges.

Això demostra que aquesta estratègia realment hi ha gent que ho ha aconseguit fer funcionar.

4 METODOLOGIA



Figura 5: Metodologia Scrum

Com a metodologia s'ha fet servir una metodologia àgil semblant a Scrum. És semblant a Scrum en el sentit de que s'han definit sprints amb una durada de 2 setmanes, on s'han anat definint les tasques a realitzar en cada iteració per a anar incloent o modificant noves funcionalitats. La raó per la qual

s'ha triat implementar una metodologia àgil, es debut a que és més flexible, cosa que permet compaginar els estudis i la feina amb el treball amb més facilitat, gracies a que no estan totes les dates de cada objectiu fixades, ademès que també gracies a que la durada dels sprints es curta, permet canviar fàcilment la planificació per a incloure nous canvis.

Per a definir el contingut dels nous sprints i fer un sprint review de com ha anat el sprint anterior, s'han tingut reunions amb el tutor del projecte, Dimosthenis Karatzas una vegada a la setmana.

Com a eina per a facilitar el dur a terme aquesta metodologia, s'ha fet servir trello, trello es una pagina web que et facilita un taulell, el qual s'ha utilitzat per a afegir les tasques a fer en cada Sprint.

Els sprints realitzats han sigut els següents:

Sprint 1: 16 sep -2oct	Pensar en els objectius del joc. Decidir amb quines eines poder fer-lo. Investigar com fer la detecció de text
Sprint 2: 2oct - 17oct	Aprendre a utilitzar unity i començar a implementar la escena i algun script.
Sprint 3: 16oct - 5nov	Intentar buscar i implementar noves formes de detecció de text. Fer una primera demo, encara sense poder detectar el text.
Sprint 4: 6nov - 18dec	S'ha decidit al fi quin modul de reconeixement utilitzar i decidit canviar el joc. També s'ha investigat com implementar el multijugador.
Sprint 5: 21dec-11gener	Primera versió del joc del laberint, escena dissenyada i sistema reconeixement de text incorporat al joc, el personatge es pot controlar amb ratolí
Sprint 6: 11gener- 22gener	Finalització del joc, el personatge es pot controlar amb pad, funcional en Android, incorporada escena de joc finalitzat i possibilitat de guardar les imatges.

Tal com es pot veure, s'ha dedicat la gran major part del temps, en obtenir el mòdul de reconeixement de text per a poder desenvolupar el joc, el qual ha fet que es tingués que reestructurar el projecte donat a falta de temps. Això ha sigut debut a que es tenia pensat incorporar el modul que facilita

google per a reconèixer text², donat que s'adaptava perfectament a les nostres necessitats, però ha costat unes setmanes aconseguir que el codi es pogués compilar correctament en AndroidStudio, després s'ha investigat com importar en Unity, el procés per a crear un plugin de android i importar-lo a unity, ha sigut bastant complicat de fer, pel que ha tardat també bastant de temps. Un cop s'ha aconseguit importar al final, no s'havia aconseguit que funcionés, així que s'ha tardat una setmana més en trobar on estava el problema, on al final s'ha vist que no es podia importar tal qual i seria molt difícil incorporar-ho, debut a que per a importar una activitat de Android a Unity, es necessari fer que aquella activitat incorpori la classe de Activitat de unity i el codi de la llibreria estava molt lligada al tipus de activitat compacta de android, pel qual al final no s'ha pogut incorporar.

A més, durant tot aquest temps, s'ha estat desenvolupant una primera versió del joc i s'ha estat explorant com desenvolupar el multijugador, el qual s'hauria d'haber deixat per més endavant i donat preferència a tenir decidit el imprescindible mòdul de reconeixement de text.

DISSENY DEL JOC

Estan desapareixent les paraules del món, fent així que desapareguin els objectes els quals representen aquelles paraules, sembla que la causa és un laberint que ha aparegut al mig de tokyo, pel qual gent ha sigut enviada allà a tractar de solucionar la causa, abans que el concepte de persona també desaparegui juntament amb ells. Han transcorregut 2 setmanes des de llavors i no se sap cap notícia de la gent que havia entrat, per tant el govern m'ha designat la gran missió de entrar en el laberint i tractar de salvar el món, donat que soc dels pocs que més o menys pot recordar la major part de les paraules oblidades.

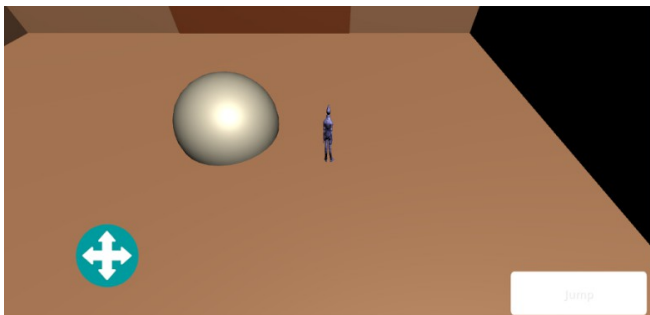


Figura 6: Sala inicial del joc, es pot veure la porta per passar de sala al fons i el pad per moure el personatge abaix a la esquerra

En el joc controlaràs un avatar, el qual aniràs explorant un laberint, cada sala del laberint està connectada amb les altres amb portes les quals, tindràs que reconèixer paraules de un format concret per a poder passar a una següent sala, així trobant la sortida cap al següent nivell, en aquesta versió del joc, només conté un sol nivell.

Per a controlar el avatar, s'ha facilitat un pad amb el qual el usuari es pot moure.

S'ha decidit per fer un laberint, debut a que semblava una temàtica entretinguda pel usuari i era una bona idea per a posar en practica tots els conceptes bàsics de unity en un nou joc.

S'ha fet servir el concepte de porta per a reconèixer imatges en escenes reals amb text, donat que es un objecte amb el qual es necessari interactuar amb la escena, així donant efecte al usuari, que està fent un progrés gracies a haber reconegut el text correcte i no efecte negativament la jugabilitat del joc, el fet de canviar a una escena completament distinta, per a reconèixer una paraula, si es en el canvi de escena.

5 DESENVOLUPAMENT

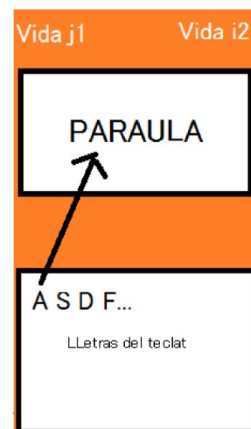


Figura 7: Primera idea de joc

La primera idea de joc que s'havia pensat, era semblant a un joc de paraules encadenades, on comences amb una lletra i tens que trobar una paraula que comenci amb ella, captant lletra per lletra tirant lletres de forma semblant a com tires pokeballs en el pokemon go al capturar un pokemon, fet amb realitat augmentada, un cop captada una paraula, tens que trobar una nova que comenci amb la lletra de la ultima paraula, el joc estava pensat en multijugador, de forma que haguessin 2 usuaris que anessin jugant de aquesta forma, al finalitzar una paraula, generaves dany a la vida del contrincant, relatiu a la longitud de la paraula i si la paraula acabava amb la primera lletra de la paraula que estava jugant el adversari, obtenies un escut que et protegia del dany de la paraula del adversari.

En la figura [8] podem veure com seria la estructura client-servidor del joc en multijugador, el servidor buscaria una partida, els usuaris es connectarien a ella, unity aporta unes variables booleanas per comprovar si el que està connectat es el usuari o el servidor, de forma que es pugui diferenciar el codi de cada una de les parts, de forma que les coses que fa el usuari en local que volem mostrar el altre usuari, haurien de ser

2: <https://developers.google.com/vision/text-overview>

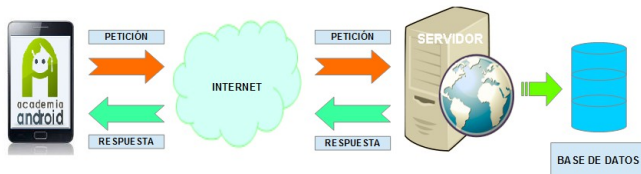


Figura 8: Estructura client-servidor per al multijugador

executades en el servidor.

Un cop tenint el servidor muntat, no seria difícil anar enviant les imatges que s'anessin captant per una crida http.

Aquesta primera idea, no es va acabar realitzant, debut a que no va es va trobar un mòdul de reconeixement de text que complís els seus requeriments I es va tenir que canviar donat a que quedava poc temps per a acabar el projecte I s'estava atraçant molt. En la secció de reconeixement de text, s'explicarà amb més profunditat perquè no es va poder trobar un mòdul adient per a aquesta idea.



Figura 9: Tecnologies utilitzades en el joc

Per tant, al final el que s'ha desenvolupat ha sigut el joc del laberint explicat en la secció de disseny, en la figura [9] podem veure les tecnologies que hem introduït al nostre joc, el joc ha sigut desenvolupat totalment en Unity, on se li ha incorporat les llibreries de Vuforia I Android, s'ha decidit utilitzar aquestes tecnologies, al veure que eren les que millor s'adaptaven per al nostre joc, donat que es complementen molt bé entre elles, a més que dona una gran facilitat a importar el nostre joc en altres plataformes en un futur, cosa que podria ser interessant importar en IOS per ampliar el número de usuaris.

Vuforia és una llibreria que dona suport per a realitat augmentada en diferents entorns, un dels quals seria Unity, té diferents mòduls amb els quals aportar un efecte de realitat augmentada, com per exemple smart terrain, reconeix objectes del món real I modelar-los en el món virtual, entre altres, però en aquest informe, ens centrarem a explicar el mòdul que s'ha fet servir, que es el de reconeixement de text.

En el procés de desenvolupament, s'han implementat els següents mòduls:

-Reconeixement de text amb Vuforia.

-El control del jugador

-Guardar el estat del joc

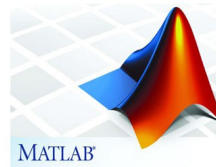
-Guardar les imatges amb els texts.

A continuació explicarem com s'ha desenvolupat cada un dels mòduls.

5.1 Mòdul de reconeixement de text



VS



Android Caffe Recognition

Figura 10: Els diferents moduls de reconeixement de text probats

Per a reconèixer text, s'ha intentat incorporar diferents llibreries abans de Vuforia, debut a que Vuforia té alguns inconvenients, però el principal que aporta, es que li costa a vegades bastant temps en reconèixer el text, debut a que la càmera tarda en agafar correctament el focus i surt el text borrós, a més de que només es capaç de reconèixer text impresa, aquestes 2 limitacions impossibilitaven la idea original de joc que s'havia pensat de joc en realitat augmentada, motiu pel qual s'ha acabat fent una versió sense incloure la interacció que aporta la realitat augmentada, però si mantenint el fet de reconèixer text.

A continuació es mostra una taula, on es pot veure els diferents mòduls de reconeixement de text que s'han intentat incorporar al nostre joc, amb les seves avantatges i inconvenients.

	Avantatges	Inconvenients
google	Pot reconèixer practicamente qualsevol paraula amb un temps curt.	Dificultat a importar en plataforma de videojocs debut a estar lligat amb la activitat compacte de android
matlab	Reconeix correctament text de impremta	No reconeix text en temps real, dificultat a incorporar en unity, s'hauria de connectar amb un servidor que pugues executar matlab.
caffe	Pot dir si hi ha text en una zona o no	No reconeix text en temps real, no sap en certesa la seva transcripció.
vuforia	Reconeix temps en	Pot tardar a vegades en

	text real, si està inclòs en el seu diccionari de paraules.	consegüir reconèixer text, només reconeix text en impremta.
--	---	---

Per tant, al final s'ha decidit incloure Vuforia, ja que es el que s'adaptava millor al nostre projecte, gracies a la seva facilitat de incloure.

Per a incorporar-lo, hem fet ús del sample que ja proporciona Vuforia pel reconeixement de text, adaptant la interfàç de forma que preguntí al usuari per un tipus de text determinat.

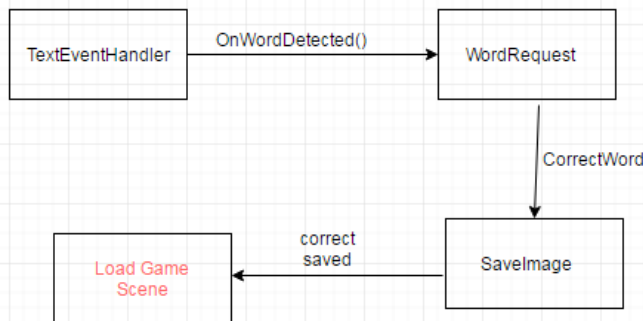


Figura 11: Diagrama representatiu del mòdul de reconeixement de text

Per a gestionar les preguntes per a diferents texts, s'ha creat un script wordRequest, el qual genera una lletra aleatòria a demanar al usuari i actualitza el text de la interfície al entrar en la escena, després es crida quan Vuforia reconeix una paraula, per verificar si conté la lletra demanada. Si es correcte, crida el mòdul de guardar la imatge i si la imatge es guarda correctament, es carrega la escena del joc de nou.

Si no es reconeix el text esperat o no es guarda la imatge, la escena segueix reconeixent text com si no hagués passat res, a vegades no es guarda la imatge correctament a la primera, però en el següent frame es guarda correctament.

5.2 El control del jugador

En el desenvolupament del joc, s'ha fet servir les primitives 3D que inclou unity per a dissenyar el escenari.

Un cop el escenari dissenyat, per a afegir el avatar del jugador, em fet ús del paquet standard Assets que proporciona unity. Aquest avatar, bé incorporat com un prefab, és a dir, un objecte guardat ja em tots els seus components necessaris per funcionar.

Per tant, el control del moviment ja venien els scripts generats, per a poder-se moure amb el teclat en la versió pc.

De aquesta forma, el jugador ja es pot moure per la pantalla, però la càmera encara no es mou, pel que surt del nostre camp de visió, per tant hem afegit un script a la càmera per a que es vagi actualitzant la seva posició mantenint-se a una distància fixa del jugador.

Un cop ja tenim el escenari creat, i el jugador es pot moure amb la camera seguint-lo, falta que pugui interactuar amb el seu entorn i que pugui guanyar la partida.

El prefab per defecte, ja conté que el avatar tingui el component rigid body, de forma que no pugui travessar objectes, i li afecti la gravetat, aquesta part de la interacció ja s'encarrega Unity i no es necessari programar-ho, però ens falta reconèixer quan el objecte amb el que col·lisionem es la porta, de forma que puguem cridar a la escena de reconeixement de text i després passar a la següent sala.

Això ho hem implementat, afegint un script en les portes, aquest script implementa el mètode OnCollisionEnter(), bàsicament significa que si 2 objectes amb component collider es toquen, s'accedeix a ell, llavors el que fem es guardar en fitxer les coordenades de la pròxima escena, sumant-li a les coordenades de la porta un bonus segons la rotació de la porta, i carrega la escena de reconeixement de text.

Tenint tot això implementat, ja només falta poder acabar la partida, per a poder guanyar, he creat una porta especial de color negre, que el seu script es semblant al anterior mencionat per una porta normal, la diferencia es que al col·lisionar, carrega la escena gameComplete.

En la escena gameComplete, tenim un text que et felicita per haver completat el joc i un botó amb el text "New Game", que al pulsarlo, borra el fitxer amb la informació de la partida anterior i crida a la pantalla de joc, per a començar una nova partida.

5.3 Guardant el estat del joc

Al canviar de escena, es perden tots els canvis que s'havien fet, per tant es necessari guardar-ho en fitxer. Per a fer això, unity utilitza el concepte de serialitzable, es a dir, només es poden guardar a fitxer dades serialitzables, això correspon a dades de tipus bàsic, com serían string, integer, bool o float, a més de les dades de tipus basic de tota una classe que sigui definida con serialitzable.

Com el que volem guardar es la posició del personatge, i això es un tipus de dada Vector3, hem tingut que generar una classe nova serialitzable, que incorpori els 3 floats a guardar, amb els quals després puguem reconstruir el Vector3 de la posició.

Un altre cosa que ha sigut necessària guardar a fitxer, ha sigut tant la informació sobre el text que apareix en les imatges que guardem, com el comptador de número de imatges que tenim guardades.

5.4 Guardant les imatges amb els texts

Per a guardar la imatge, primer de tot tenim que obtenir la imatge, donat que fins ara només teníem el string del text.

Per a obtenir la imatge, necessitem primer definir un format del pixel, el format que hem escollit ha sigut greyScale, debut a que els formats en color no els agafava correctament la càmera de Vuforia, un cop comprovat que el format afegit es correcta, agafem la imatge segons aquell format, a vegades no és capaç de recollir correctament la imatge, pel que es té que verificar que no sigui null, un cop verificat ja finalment podem guardar

una imatge correcta.

Per a guardar la imatge, el primer pas es passar-la a Texture2D, això ens permet definir un tipus de Encode, ja que sino simplement estariem guardant un array de bytes, pel qual no podríem obrir la imatge guardada correctament.

Al passar a Texture2d, ja podem cridar al metode System.IO.File.WriteAllBytes, que agafa com a paràmetre la path a on guardar i el array de bytes codificat i ja tenim la imatge guardada correctament. El encode que hem utilitzat ha sigut el de PNG, en el nom de la path incluem un contador per a que no sobreescrigui la imatge anterior i generi una nova imatge amb un número diferent, que tal com s'ha explicat anteriorment es guarda posteriorment a fitxer.

6 RESULTATS

Per a avaluar els resultats, s'ha generat el apk del joc i s'ha compartit a un petit grup de 3 usuaris i s'els ha fet un breu user testing individual.

Com a preguntes introductòries del user testing s'han fet servir:

1-Quantes hores sols utilitzar el mòbil?

2-Has jugat el pokemon go?

3-A part del pokemon go, has jugat algún altre joc de realitat augmentada? Quins?

1 usuari jugava 4h, un altre 1h i un altre 2h al dia.

Tots els usuaris han jugat al pokemon Go.

1 usuari ha jugat altres jocs de realitat augmentada, entre ells el joc de ingress, que es semblant al pokemon Go.

A continuació s'els ha explicat que estaven en una sessió de user testing, que no es preocupessin de si s'equivocaven, ja que ells no es poden equivocar, més aviat seria que la aplicació esta mal feta i que es necessari que anés donant informació de tot el que anava fent en veu alta. Després s'els li ha donat el joc i mentres els jugava, s'els insistia de que anessin parlant, preguntantlis sobre el que anaven fent.

Després de acabar els user testings i apuntar tota la informació que s'ha extret dels usuaris, s'ha fet una llista dels canvis que s'ha considerat importants a canviar, dels quals molts d'ells, debut a centrar-se més en la programació que en el disseny, no s'havien contemplat.

Els resultats del user testing serian els següents, qualificats segons la reacció del usuari i la pròpia opinió:

Crític:

-El avatar es massa petit en proporció amb la resta del escenari.

-El avatar es mou massa lent.

-La lletra a buscar no la detecta si està en majúscula.

-S'ha detectat un bug en el qual, si estàs en la escena de reconeixement de text i tancàs la finestra de la aplicació i la tornes a obrir, saltes a la següent sala sense necessitat de

reconeixer el text.

-Al no avisar que només es reconeix paraules en anglés, sembla que ha dificultat molt i per tant frustrat als usuaris, el fet de aconseguir detectar una paraula correcta.

No crític:

-Tarda massa a canviar entre escenes.

-A vegades costa una mica detectar les paraules.

-Al haver poques textures, era fàcil quedar-te en alguna zona on no hagués res en el al voltant, pel que costava orientar-se.

-Sembla que no agrada massa el fet de que el personatge faci un petit salt desde el aire al apareixer en la sala.

Per un altre banda, en la avaluació del text recognition, es podria dir el següent:

-Només reconeix el text si està en paral·lel amb la càmera.

-Només detecta lletra impresa.

-A vegades costa bastant que reconegui el text, ja que apareix borrós en la càmera.

-No detecta lletres soltes.

-Detecta les paraules fora de la zona de detectar el text.

-Si les paraules son llargues, li costa més detectar-les.

3 CONCLUSIÓ

La idea inicial de joc, ha tingut que canviar debut a que no s'ha trobat un detector de text adient pel que s'esperava fer i no es podia endarrerir més tot el projecte debut a això. Per tant, el joc final no ha pogut complir tots els objectius esperats, ha faltat implementar el mode multijugador, el qual no s'ha implementat per 2 raons, la primera es que després de dedicar-li bastant de temps investigant com afegir-lo, s'ha vist que era bastant més complex del que es pensava i faltaria temps per a incloure-ho, la segona raó és perquè el nou model de joc, no estava pensat per jugar en mode multijugador.

Pel que respecta a la resta dels objectius, s'ha aconseguit un joc funcional en android amb un reconeixement de text que si bé té les seves limitacions, es totalment funcional pel model final de joc.

També s'ha assolit tenir una petita base de dades amb les imatges i meta informació, que si bé ara mateix estaria en local, guardat en el mateix dispositiu, seria fàcilment importable, a una base de dades real.

Per altre banda, tenint en compte la avaluació dels resultats, el joc ha sigut el suficient intuitiu per a que els usuaris sàpiguem com jugar-hi i han pogut acabar el joc en un temps entre 10-15 minuts i la major part dels problemes que s'ha trobat al joc, han sigut debut a que no s'ha li ha donat el suficient temps en dissenyar correctament la interfície.

També, podríem dir que tenint en compte el tipus de joc que es volia crear, un laberint amb el qual els usuaris es

perdessin, els costés molt arribar a la meta i es frustressin en el camí, en això ha tingut tot un èxit i es podria aprofitar en un futur, donat que el fet de que, un joc passable en 2minuts es passí en 10-15 minuts, òbviament això ha tingut que donar-li un factor de dificultat extra bastant elevat, tenint en compte que el joc no suposa cap altre obstacle que tenir que reconèixer text.

En resum, s'ha aconseguit tenir un joc funcional, on puguem navegar pel mon, anar detectant imatges per anar avançant, les quals guardem en una base de dades, la qual podria ser utilitzada per a futures recerques.

El joc encara està en una versió molt inicial, segons la experiència dels usuaris, sembla que encara falta canviar moltes coses, el qual podria augmentar molt la seva qualitat i poder arribar a tenir èxit si es dediques més temps.

En quant a referències per a un futur, gracies a aquest projecte, he après totes les coses essencials sobre fer jocs amb unity, el qual em serà de una gran ajuda, debut a que el meu objectiu es ser un desenvolupador de videojocs i estava pensant fer un futur master de videojocs.

També he après la importància de planificar correctament un projecte i tenir clar totes les eines i com es desenvolupar totes les fases de un projecte, abans de començar a programar, per a no tenir que treballar molt més del compte.

Espero que la experiència de aquest treball m'ajudi molt en un futur i els meus nous projectes arribin a tenir molt més èxit.

AGRAÏMENTS

Agraeixo al tutor Dimosthenis Karatzas tota la ajuda que m'ha proporcionat, a més de la paciència que ha tingut i els ànims que m'ha donat.

També agraeixo a la meua família, per per tot el suport que m'ha donat durant tota la carrera.

BIBLIOGRAFIA

- [1] Muhammad, Amgad. *OpenCV Android Programming By Example*. Packt Publishing Ltd, 2015.
- [2] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, D. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, VR. Chandrasekhar, A. Lu, F. Shafait, S. Uchida, E. Valveny, "ICDAR 2015 Robust Reading Competition," 13th International Conference on Document Analysis and Recognition (ICDAR).
- [3] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Gomez, S. Robles, J. Mas, D. Fernandez, J. Almazan, L.P. de las Heras, "ICDAR 2013 Robust Reading Competition," in Proc. IEEE Int'l Conf. Document Analysis and Recognition, pp. 1484-1493, 2013.
- [4] Q. Ye, D. Doermann, Text detection and recognition in imagery: A survey, *Pattern Analysis and Machine Intelligence*, IEEE Transactions on 37 (7) (2015) 1480–1500. 5
- [5] Luis von Ahn and Laura Dabbish. Labeling images with a computer game. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '04, pages 319–326, New York, NY, USA, 2004. ACM.
- [6] The rise of Crowdsourcing. (2017, January 21). Retrieved from <https://www.wired.com/2006/06/crowds/>
- [7] Luis von Ahn: Massive-Scale Online Collaboration. (2016, December 28). Retrieved from

https://www.ted.com/talks/luis_von_ahn_massive_scale_online_collaboration

- [8] Von Ahn, Luis, Ruoran Liu, and Manuel Blum. "Peekaboom: a game for locating objects in images." *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006.
- [9] Von Ahn, Luis, Mihir Kedia, and Manuel Blum. "Verbosity: a game for collecting common-sense facts." *Proceedings of the SIGCHI conference on Human Factors in computing systems*. ACM, 2006.
- [10] González, C., et al. "Realidad aumentada. Un enfoque práctico con ARToolkit y Blender." (2013).

